of Computational Linguistics (NODALIDA-2007). University of Tartu, Tartu. May 24-26, 2007. - Tartu: University of Tartu, 2007. Pp. 372-379.

Evaluating Evaluation Measures

Ines Rehbein
NCLT
School of Computing, DCU,
Dublin, Ireland
irehbein@computing.dcu.ie

Josef van Genabith
NCLT,
School of Computing, DCU,
Dublin, Ireland
josef@computing.dcu.ie

Abstract

This paper presents a thorough examination of the validity of three evaluation measures on parser output. We assess parser performance of an unlexicalised probabilistic parser trained on two German treebanks with different annotation schemes and evaluate parsing results using the PARSEVAL metric, the Leaf-Ancestor metric and a dependency-based evaluation. We reject the claim that the TüBa-D/Z annotation scheme is more adequate then the TIGER scheme for PCFG parsing and show that PARSEVAL should not be used to compare parser performance for parsers trained on treebanks with different annotation schemes. An analysis of specific error types indicates that the dependency-based evaluation is most appropriate to reflect parse quality.

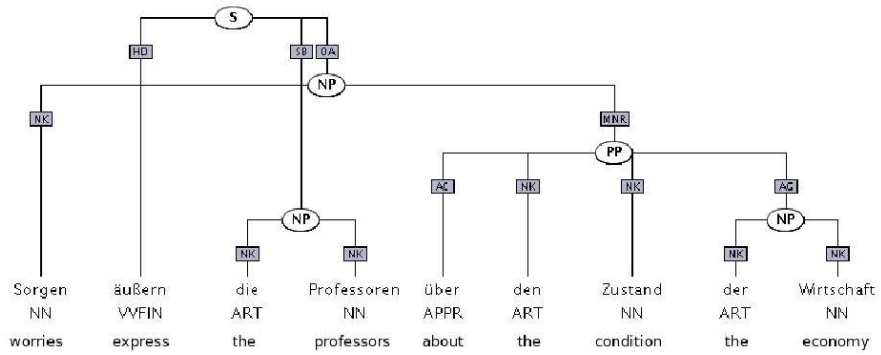
1 Introduction

The evaluation of parsing results is a crucial topic in NLP. Despite severe criticism for PCFG parsing the PARSEVAL metric is still the standard evaluation measure. PARSEVAL has been criticised for not representing 'real' parser quality (Carroll et al., 1998; Brisco et al., 2002; Sampson et al., 2003).

Recent studies investigating the impact of different treebank annotation schemes on unlexicalised probabilistic parsing of German (Kübler, 2005; Kübler et al., 2006; Maier, 2006) have been using the PARSEVAL metric for evaluation. Results (labelled bracketing f-score) are about 16% higher for a parser trained on the TüBa-D/Z treebank (Telljohann et al., 2004) than for a parser trained on the NEGRA treebank (Skut et al., 1997). Maier (2006) takes that as evidence that the NEGRA annotation scheme is less adequate for PCFG parsing, while a parser trained on the TüBa-D/Z yields PARSEVAL results in the same range as a parser trained on the English Penn-II treebank (Kübler et al., 2006). These results are based on the assumption that PARSEVAL is an appropriate measure for comparing parser performance of a PCFG parser trained on treebanks with different annotation schemes.

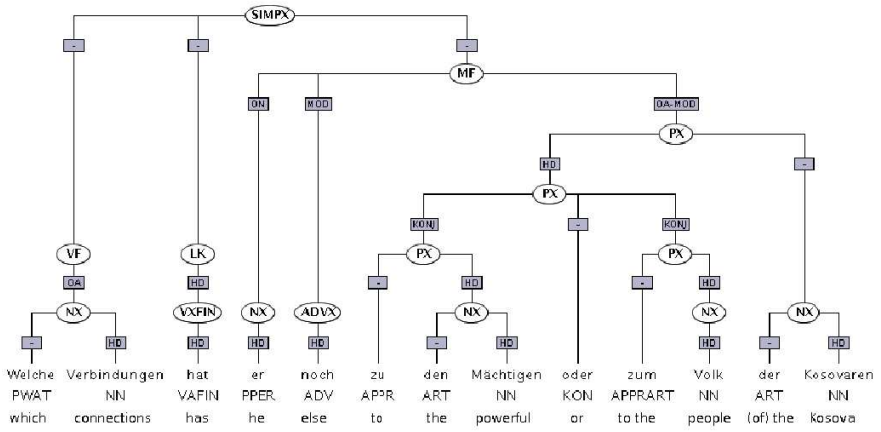
This paper presents parsing experiments with the PCFG parser BitPar (Schmid, 2004) trained on two German treebanks. The treebanks contain text from the same domain, namely two German daily newspapers, but differ considerably with regard to their annotation schemes. We score parsing results using three different evaluation measures and show that the PARSEVAL results do not correlate with the results of the other metrics. An analysis of specific error types shows the differences between the three measures. Our results indicate that dependency-based evaluation is most appropriate to compare parser output for parsers trained on different treebank annotation schemes.

Section 2 describes the main features of the two German treebanks, and Section 3 gives an overview over the metrics used for evaluation. Section 4 presents the parsing experiments. In Section 5 we describe the behaviour of the different evaluation metrics for specific error types. Section 6 concludes.



"The professors express their concerns about the state of the economy."

Figure 1: TIGER Treebank tree



"What connections does he still have with those in power or with the people of Kosovo."

Figure 2: TüBa-D/Z Treebank tree

2 TIGER and TüBa-D/Z

The two German treebanks used in our experiments are the TIGER Treebank (Release 2) and the Tüba-D/Z (Release 2). The TüBa-D/Z consists of approximately 22 000 sentences, while the TIGER Treebank is much larger with more than 50 000 sentences. TIGER is based on and extends the NEGRA data and annotation scheme. Both treebanks contain German newspaper text (Frankfurter Rundschau for TIGER and 'die tageszeitung' (taz) for TüBa-D/Z) and are annotated with phrase structure and dependency (functional) information. Both treebanks use the Stuttgart Tübingen POS Tag Set (Schiller et al., 1995). TIGER uses 49 different grammatical function labels, while the TüBa-D/Z utilises only 36 function labels. For the encoding of phrasal node categories the TüBa-D/Z uses 30 different categories, the TIGER Treebank uses a set of 27 category labels.

Other major differences between the two treebanks are: in the TIGER Treebank long distance dependencies are expressed through crossing branches (Figure 1), while in the TüBa-D/Z the same phenomenon is expressed with the help of grammatical function labels (Figure 2). The annotation in the TIGER Treebank is rather flat and allows no unary branching, whereas the nodes in the TüBa-D/Z do contain unary branches and a more hierarchical structure, resulting in a much deeper tree structure than the trees in the TIGER Treebank. This results in an average higher number of nodes per sentence for the TüBa-D/Z. Table 1 shows the differences in the ratio of nodes for the TIGER Treebank and the TüBa-D/Z.

	phrasal nodes/sent	phrasal nodes/word	words /sent
TIGER	8.29	0.47	17.60
TüBa-D/Z	20.69	1.20	17.27

Table 1: Average number of phrasal nodes/words in TIGER and TüBa-D/Z

Figures 1 and 2 also illustrate the different annotation of PPs in both annotation schemes. In the TIGER Treebank the internal structure of the PP is flat and the adjective and noun inside the PP are directly attached to the PP, while the TüBa-D/Z is more hierarchical and inserts an additional NP node. Crossing branches show the long distance dependency between the PP and the noun *Sorgen* (worries) in the TIGER tree, while in the TüBa-D/Z the node label OA-MOD encodes the information that the PP modifies the accusative object *Verbindungen* (connections).

Another major difference is the annotation of topological fields in the style of Drach (1937) in the TüBa-D/Z. The model captures German word order, which accepts three possible sentence configurations (verb first, verb second and verb last), by providing fields like the initial field (VF), the middle field (MF) and the final field (NF). The fields are positioned relative to the verb, which can fill in the left (LK) or the right sentence bracket (VC). The ordering of topological fields is determined by syntactic constraints.

2.1 Differences between TIGER and NEGRA

To date, most PCFG parsing for German has been done using the NEGRA corpus as a training resource. The annotation scheme of the TIGER Treebank is based on the NEGRA annotation scheme, but it also employs some important extensions, which include the annotation of verb-subcategorisation, appositions and parentheses, coordinations and the encoding of proper nouns (Brants and Hansen, 2002).

3 The Evaluation Measures

The three evaluation metrics used in our experiments are:

- the PARSEVAL metric (**PV**)
- the Leaf-Ancestor metric (**LA**)
- a dependency-based evaluation (**DB**)

Below we demonstrate the differences between the three evaluation measures, using Sentence 4 from the TIGER test set as an example:

(1) Die Regierung rief zum weltweiten Kampf
the government called to the worldwide fight
gegen Terror auf.
against terror up
"The government called for a worldwide war against terror."

3.1 PARSEVAL

PARSEVAL checks label and wordspan identity in parser output compared to the original treebank trees, but neither weights results, differentiating between linguistically more or less severe errors, nor does it give credit to constituents where the syntactic categories have been recognised correctly but the phrase boundary is slightly wrong.

Figure 3 shows the gold tree for our example sentence (1). In the parser output the second PP was incorrectly attached to the sentence level (Figure 4) instead of being attached to the noun inside the PP.

```
(TOP
    (S
      (NP
          (ART Die [the] )
          (NN Regierung [government] )
      )
      (VVFIN rief)
      (PP
          (APPR zum [to the] )
          (ADJA weltweiten [worldwide] )
          (NN Kampf [fight] )
          (PP
              (APPR gegen [against] )
              (NN Terror [terror] )
          )
      )
      (PTKVZ auf)
    )
    ($. .)
)
```

Die Regierung rief zum weltweiten Kampf gegen terror auf.
"The government called for a worldwide war against terror."

Figure 3: Gold tree for example (1)

PARSEVAL counts 4 (out of 5) matching brackets, which results in a precision and recall of 80.00% respectively.

```
(TOP
    (S
      (NP
          (ART Die [the] )
          (NN Regierung [government] )
      )
      (VVFIN rief)
      (PP
          (APPR zum [to the] )
          (ADJA weltweiten [worldwide] )
          (NN Kampf [fight] )
      )
      (PP
          (APPR gegen [against] )
          (NN Terror [terror] )
      )
      (PTKVZ auf)
    )
    ($. .)
)
```

Die Regierung rief zum weltweiten Kampf gegen terror auf.
"The government called for a worldwide war against terror."

Figure 4: Parser output tree for example (1)

3.2 Leaf-Ancestor

LA (Sampson et al., 2003) measures the similarity between the path from each terminal node in the parser output tree to the root node and the corresponding path in the gold tree. The path consists of the sequence of node labels between the terminal node and the root node, and the similarity of two paths is calculated by using the Levenshtein distance (Levenshtein, 1966).

For each terminal node in the parser output the sequence of node labels between the terminal node and the root node is compared to the same path in the gold tree. This results in an evaluation result for string similarity for each terminal node. The score for the whole tree is the average of the values for all terminals in the tree. Figure 5 shows LA evaluation results for example (1). The numbers in the left column give the LA scores for each terminal node, which results in an average score of 0.963 for the whole sentence.

Phrase boundaries are taken into consideration, in order to distinguish between paths like the one for *zum* (`[PP S TOP`) and the one for *weltweiten* (`PP S TOP`) in Figure 5. The LA metric does this as follows: for each terminal at the beginning of a phrase LA looks for the highest non-terminal node governing the phrase which also starts with the terminal, and inserts a left boundary marker before the categorial label of the non-terminal node, if the phrase starts with the terminal node. For *Die* [the] the highest non-terminal governing node is TOP. As the sentence starts with *Die*, a left boundary marker is inserted before the TOP node. For *zum* the highest governing non-terminal node which starts with the word *zum* is the PP, therefore the left boundary marker is inserted before the PP node.

```
1.000 Die          NP  S [    TOP  : NP  S [  TOP
1.000 Regierung    NP  ] S    TOP  : NP  ] S  TOP
1.000 rief             S    TOP  : S   TOP
1.000 zum          [   PP S    TOP  : [   PP S  TOP
1.000 weltweiten       PP S    TOP  : PP  S  TOP
0.857 Kampf            PP S    TOP  : PP  ] S  TOP
0.889 gegen        [ PP PP S    TOP  : [   PP S  TOP
0.889 Terror       PP PP ] S    TOP  : PP  ] S  TOP
1.000 auf              S  ]     TOP  : S   ] TOP
1.000 .                   TOP ]      : TOP ]
```

Sentence 1: avg. 0.963

Figure 5: LA result for example (1)

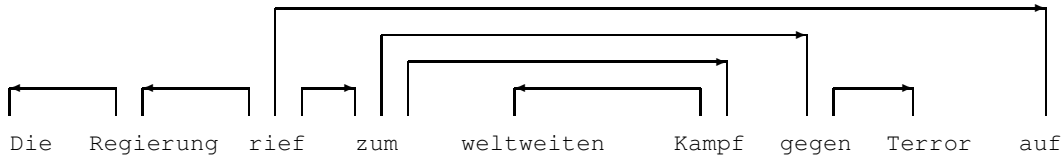
Additionally, LA looks at each terminal node at the end of a phrase and inserts a right boundary marker after the label of the highest non-terminal node of the phrase ending with the terminal node. In the gold tree of our example the terminal node *Kampf* [fight] is not the final node of the PP. Due to the PP attachment error in the parser output tree, on the other hand, *Kampf* [fight] is in a phrase-final position, with the PP as the highest non-terminal node governing the terminal. Therefore a right boundary marker is inserted after the PP node in the path of the parser output, which results in a score of 0.889 for path similarity between gold tree and parser output.

The average result for the whole sentence is 0.963, while a perfect sentence would get a score of 1. If LA encounters a mismatch between the words in the gold tree and the parser output, it simply stops without returning a result for the whole sentence.

3.3 Dependency-Based Evaluation

The dependency-based evaluation used in the experiments follows the method of Lin (1998) and Kübler et al. (2002), converting the original treebank trees and the parser output into dependency relationships of the form WORD POS HEAD. Functional labels have been omitted for parsing, therefore the dependencies do not comprise functional information.

Figure 6 shows the dependency relations for example (1), indicated by arrows. Converted into a



"The government called for a worldwide war against terror."

Figure 6: Dependency relations for example (1)

WORD POS HEAD triple format the dependency tree looks as follows (Table 2).

WORD		POS	HEAD
Die	[the]	ART	Regierung
Regierung	[government]	NN	rief
rief	[called]	VVFIN	-
zum	[to the]	APPRART	rief
weltweiten	[worldwide]	ADJA	Kampf
Kampf	[fight]	NN	zum
gegen	[against]	APPR	zum
Terror	[terror]	NN	gegen
auf	[up]	PTKVZ	rief

Table 2: Gold dependency triples for example (1)

The PP attachment error in the parser output leads to an error in the dependency triples, incorrectly assigning *rief* [called] as the head of *gegen* [against] (Table 3), while in the gold triples the PP *gegen Terror* [against terror] is a dependent of the preposition *zum* [to the].

WORD		POS	HEAD
gegen	[against]	APPR	rief

Table 3: Error in parser output dependency triples

For our example we get a precision and recall of 88.89 respectively. Following Lin (1998), our algorithm computes precision and recall:

- **Precision**: the percentage of dependency relationships in the parser output that are also found in the gold triples
- **Recall**: the percentage of dependency relationships in the gold triples that are also found in the parser output triples.

We assessed the quality of the automatic dependency conversion methodology by converting the 1024 original trees from each of our test sets into dependency relations, using the functional labels in the original trees to determine the dependencies. We then removed all functional information from the trees and converted the stripped trees into dependencies, using heuristics to find the head. We evaluated the dependencies for the stripped gold trees against the dependencies for the original gold trees including functional labels and obtained an f-score of 99.65% for TIGER and 99.13% for the TüBa-D/Z dependencies. This shows that the conversion is reliable and not unduly biased to either the TIGER or TüBa-D/Z annotation schemes.

4 Experimental Setup

For the experiments we trained the PCFG parser BitPar (Schmid, 2004) on the TIGER treebank and the TüBa-D/Z. The training sets for each treebank contain 21067 sentences, while the test sets include 1024 sentences each. To allow a meaningful comparison of parsing results we selected sentences comparable with regard to sentence length, syntactic structure and complexity from both treebanks for our test sets. This resulted in an average sentence length of 14.5 for the TIGER test set and of 14.7 for the TüBa-D/Z. Before extracting the grammars we inserted a virtual root node and resolved the crossing branches in the TIGER treebank by attaching the non-head child nodes higher up in the tree. After this preprocessing step we extracted an unlexicalised PCFG from each of our training sets. We parsed our test sets with the extracted grammars, using raw text as parser input.

5 Results

Table 4 shows the evaluation results for the different metrics.[1] PARSEVAL shows higher results for

[1] PARSEVAL results report *labelled* precision and recall.

precision and recall for the TüBa-D/Z. For DB evaluation the parser trained on the TIGER training set achieves about 7% higher results for precision and recall than the parser trained on the TüBa-D/Z. The LA results are much closer to each other, but also show better results for the TIGER parse trees.

	PARSEVAL		Dependencies		LA
	Prec	Rec	Prec	Rec	Avg.
TIGER	81.21	81.04	85.78	85.79	0.9388
TüBa	87.24	83.77	78.63	78.61	0.9258

Table 4: Parsing results for three evaluation metrics

Comparing the f-score learning curves in the three metrics shows that for PARSEVAL the gap between TIGER and TüBa-D/Z is consistent throughout the whole training process. But while during the first stages of training the difference in results adds up to around 12%, the gap becomes smaller with more training data. When trained on 90-100% of the training data, the difference in f-scores decreases to around 5% (Figure 7).

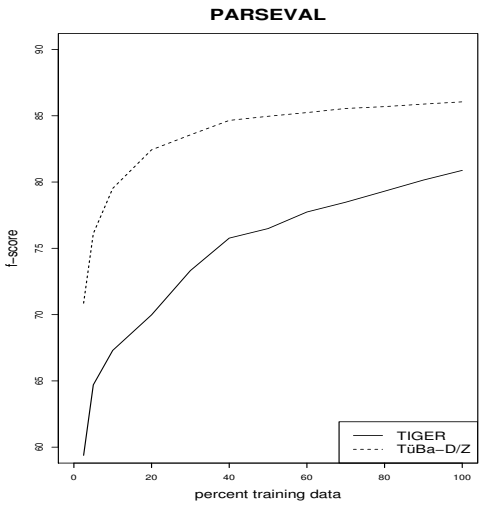


Figure 7: F-score learning curves for TIGER and TüBa-D/Z (PARSEVAL)

The LA learning curve shows an advantage for TüBa-D/Z during the first stages of training. When training the parser on less than 20% of the training data, the TüBa-D/Z-trained grammar yields better results. Training the parser on more than 50% of the sentences in the training set reverses the picture: while the f-score for TüBa-D-Z does not seem to improve further, the TIGER results clearly show an ongoing learning effect (Figure 8).

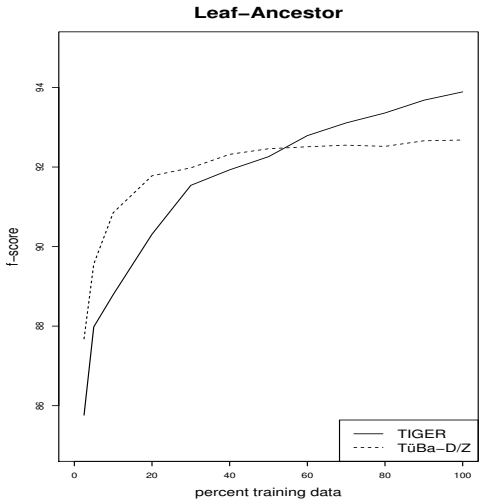


Figure 8: F-score learning curves for TIGER and TüBa-D/Z (Leaf-Ancestor)

The learning curve for the dependency-based evaluation (Figure 9) shows a similar tendency. While the TüBa-D/Z yields better results when trained on a small amount of training data only, and from more than 20% of the training set onwards only shows a moderate increase, the f-score for TIGER improves faster and shows an advantage of more than 6% over the TüBa-D/Z f-score when trained on the whole training set.

The wide difference between the results raises the question, which of the metrics is the most adequate for judging parser quality. The next section approaches this question by looking at the behaviour of the different metrics with regard to specific error types.

5.1 Part-of-Speech Errors

Parse trees yielding 100% precision and recall for PARSEVAL and 100% for LA, but failing to get 100% precision and recall for the DB evaluation, often contain POS errors. In most cases the parser assigned a noun tag instead of a proper name, an adjective tag instead of a cardinal number, or mixed up attributive adjectives with predicative adjectives. These error types are attested in 32 sentences in the TIGER and in 23 sentences in the TüBa-D/Z test set.

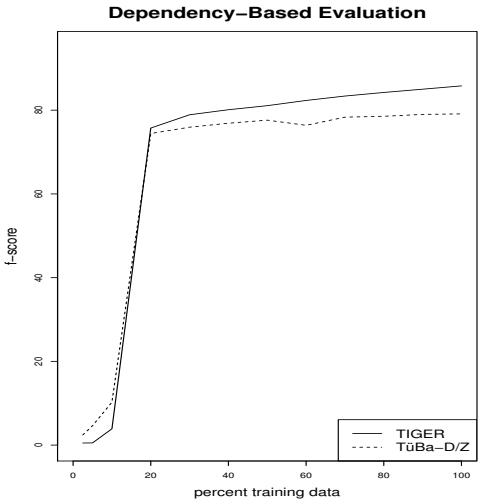


Figure 9: F-score learning curves for TIGER and TüBa-D/Z (Dependencies)

5.2 Missing Nodes / Additional Nodes

Parse trees achieving 100% precision and recall for DB evaluation, but not for the PARSEVAL and LA metric mostly lack a non-terminal node such as a proper name node enclosing an NP, a multi-token number for the TIGER treebank or the *Nachfeld* (final field) for the TüBa-D/Z. This applies to 23 sentences in the TIGER test set and to 29 sentences in the TüBa-D/Z test set. In the parser output of the parser trained on the TIGER treebank there are also sentences which show additional categorial nodes not present in the gold trees, such as prepositional phrases enclosing a pronominal adverb, adverbial phrases or adjectival phrases. Both the missing and the additional nodes do not translate into dependency errors as the dependencies for the trees can be extracted correctly. Nonetheless they lead to a significant decrease in precision and recall for the PARSEVAL scores and, to a lesser extent, also for the LA scores.

5.3 PP Attachment Errors

Parse trees with attachment errors often get reasonable results for the PARSEVAL metric but only show mediocre scores for DB evaluation. We demonstrate this for two sentences with PP attachment errors from the TIGER test set (Figure 1) and the TüBa-D/Z (Figure 2).

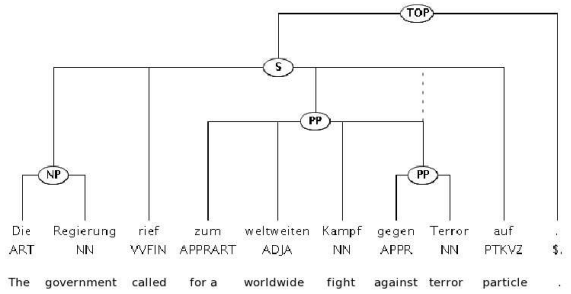


Figure 10: TIGER (dotted line: parser output)

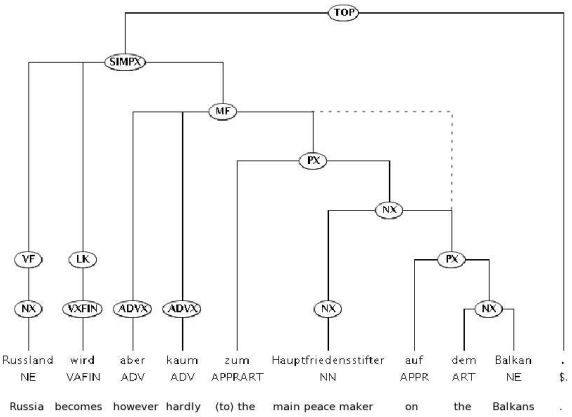


Figure 11: TüBa-D/Z (dotted line: parser output)

In the gold trees one of the PPs is a child respectively grandchild of the other PP, while in the parser outputs both PPs are mis-attached to the same mother node (dotted lines). Table 5 shows the evaluation results for the two sentences.

	PARSEVAL		Dependencies		LA
	Prec	Rec	Prec	Rec	Avg.
TIGER	80.00	80.00	88.89	88.89	0.963
TüBa	92.31	85.71	88.89	88.89	0.935

Table 5: Parsing results for PP attachment error

Despite the similarity of the two trees concerning sentence length and syntactic complexity PARSEVAL yields strongly different results for the TIGER and the TüBa-D/Z parser output. The LA scores are much closer, giving better results for the TIGER parse tree, while the PARSEVAL results are clearly in favour of the TüBa-D/Z tree. The difference be-

tween the PARSEVAL results for two comparable trees is caused by the higher ratio of nodes per words in the TüBa-D/Z annotation scheme. For the TIGER tree the parser is able to match 4 out of 5 brackets which yields a recall of 4/5 = 80%. For the TüBa-D/Z the parser correctly identifies 12 out of 14 brackets in the gold tree and therefore achieves a recall value of 12/14 = 92.31%. The dependency-based evaluation gives identical results for the two sentences, which is what linguistic intuition would ask for.

6 Conclusions

In this paper we rejected the claim that the German TüBa-D/Z is more appropriate for PCFG parsing than the TIGER treebank. We showed that the PARSEVAL metric cannot be used to compare parser output from parsers trained on different treebanks, because it favours annotation schemes with a high ratio of nodes per word. We have also shown that PARSEVAL results do not correlate with other evaluation measures like the Leaf-Ancestor metric or a dependency-based evaluation, and that the results of a dependency-based evaluation best reflect the linguistic notion of a good parse.

References

Dipper, S., T. Brants, W. Lezius, O. Plaehn, and G. Smith. 2001. The TIGER Treebank. *In Third Workshop on Linguistically Interpreted Corpora LINC-2001*, Leuven, Belgium.

Brants, Sabine, and Silvia Hansen 2002. Developments in the TIGER Annotation Scheme and their Realization in the Corpus. *In Proceedings of the Third Conference on Language Resources and Evaluation (LREC 2002)*, pp. 1643-1649. Las Palmas, Canary Islands.

Briscoe, E. J., J. A. Carroll, and A. Copestake. 2002. Relational evaluation schemes. *In Proceedings Workshop 'Beyond Parseval - towards improved evaluation measures for parsing systems', 3rd International Conference on Language Resources and Evaluation*, pp. 4-38. Las Palmas, Canary Islands.

Carroll, J., E. Briscoe and A. Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. *In Proceedings of the 1st International Conference on Language Resources and Evaluation*, Granada, Spain. 447-454.

Dipper, S., T. Brants, W. Lezius, O. Plaehn, and G. Smith. 2001. The TIGER Treebank. *In Third Workshop on Linguistically Interpreted Corpora LINC-2001*, Leuven, Belgium.

Drach, Erich. 1937. *Grundgedanken der Deutschen Satzlehre.* Frankfurt/M.

Kübler, Sandra, and Heike Telljohann. 2002. Towards a Dependency-Oriented Evaluation for Partial Parsing. *In Proceedings of Beyond PARSEVAL – Towards Improved Evaluation Measures for Parsing Systems (LREC 2002 Workshop)*, Las Palmas, Gran Canaria.

Kübler, Sandra. 2005. How Do Treebank Annotation Schemes Influence Parsing Results? Or How Not to Compare Apples And Oranges. *In Proceedings of RANLP 2005*, Borovets, Bulgaria.

Kübler, Sandra, Erhard Hinrichs, and Wolfgang Maier. 2006. Is it Really that Difficult to Parse German? *In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP*, Sydney, Australia.

Levenshtein, V. I. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics - Doklady*, 10.707-10 (translation of Russian original published in 1965).

Lin, Dekang. 1998. Dependency-based Evaluation of MINIPAR. *Workshop on the Evaluation of Parsing Systems*, Granada, Spain.

Maier, Wolfgang. 2006. Annotation Schemes and their Influence on Parsing Results. *In Proceedings of the COLING/ACL 2006 Student Research Workshop)*, Sydney, Australia.

Sampson, Geoffrey, and Anna Babarczy. 2003. A test of the leaf-ancestor metric for parse accuracy. *Natural Language Engineering*, 9 (4):365-380.

Schmid, Helmut. 2004. Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vectors. *In Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland.

Skut, Wojciech, Brigitte Krann, Thorsten Brants, and Hans Uszkoreit. 1997. An annotation scheme for free word order languages. *In Proceedings of ANLP 1997*, Washington, D.C.

Telljohann, Heike, Erhard W. Hinrichs, Sandra Kübler, and Heike Zinsmeister. 2005. *Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z).* Seminar für Sprachwissenschaft, Universität Tübingen, Germany.

Schiller, Anne, Simone Teufel, and Christine Thielen. 1995. *Guidelines fr das Tagging deutscher Textcorpora mit STTS.* Technical Report, IMS-CL, University Stuttgart.